ELSEVIER

Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE @ DIRECT®

Theoretical Computer Science 306 (2003) 39–53

Theoretical Computer Science

www.elsevier.com/locate/tcs

Function-dependent teams in eco-grammar systems

Dietmar Wätjen

Institut für Theoretische Informatik, Technische Universität Braunschweig, Postfach 3329, D-38023 Braunschweig, Germany

Received 18 April 2001; received in revised form 5 February 2003; accepted 7 February 2003
Communicated by A. Salomaa

Abstract

In this paper, we investigate simple eco-grammar systems with n agents. The number of agents which are active at each derivation step depends on the number of steps which have already been carried out since the beginning of the development. This dependency is expressed by a function f. For each pair of n and f, corresponding language families are defined. These families are compared with each other according to the different values of n and f.

Keywords: Formal languages; Lindenmayer systems; Eco-grammar systems

1. Introduction

Eco-grammar systems have been introduced in [3] to model the interaction between an eco-system and the organisms living in it. An eco-system can be seen as a special multi-agent system where the agents not only interact with each other but also with their common shared environment. In the approach given in [3,4], an eco-grammar system consists of an environment represented by a Lindenmayer system which acts in parallel on a string, that is, on the environmental state, and several agents which, by applying one of their action rules, change the actual environmental state at exactly one position. In the original model, the choice of an acting rule of an agent usually depends on the actual state of the environment.

In this paper we consider simple eco-grammar systems that is systems where the agents, independently of the actual state, can execute all possible actions on the

E-mail address: waetjen@iti.cs.tu-bs.de (D. Wätjen).

doi:10.1016/S0304-3975(03)00208-1

environment. Furthermore, we assume that there exist teams of agents. Teams of agents in simple eco-grammar systems have already been considered in [1,2,5], or [6]. In such a case, the behaviour of an eco-grammar system depends on the total number of its agents and on the number of agents in an active team. In [5], there were investigated teams with a fixed size. In [6], there are considered dynamical teams which are formed according to the actual capability of activating the agents. In this paper, we allow different sizes of the teams at different steps of the development where the size depends on the number of derivation steps which have already been carried out since the beginning of the development with the initial state. Such a functional dependency is formally specified by a function $f:\mathbb{N}\to\{0,\dots,n\}$, where n is the total number of the system's agents. A similar functional dependency, although in another context, has been investigated for function-limited 0L systems (see [7,10]).

In Section 2, we give the definition of simple eco-grammar systems with n agents and with function-dependent teams. If the function dependency is given by $f:\mathbb{N}\to\{0,\dots,n\}$, then we distinguish between the f-mode and $\leqslant f$-mode of derivation. During the kth step of the development, in the first case exactly $f(k)$ agents form a team, in the second case at most $f(k)$. Corresponding languages and language families are defined. Furthermore, examples are given which are used throughout the paper. These examples are different from those of [5]. First results are presented which concern the influence of the computability of the function f.

A lot of properties of the special languages of the examples are investigated in Section 3. They are used in Section 4 for different comparability results. We see that the family of 0L languages is included (usually with the exception of a finite set of words) in the language families defined by simple eco-grammar systems with n agents according to the f- or $\leqslant f$-team mode. Most of these inclusions are strict. Besides some incomparability results, we also get infinite hierarchies of some of these language families. As special cases, we can derive the results from Section 4 and 5 of [5] with the exception of Theorem 4.1(iii) of that paper.

2. Definitions, examples and first results

In the following, we denote by $\mathbb{N}$ the set of all natural numbers (where $0\notin\mathbb{N}$). Then $\mathbb{N}_0=\mathbb{N}\cup\{0\}$. For an alphabet V and $a\in V$, $V'\subseteq V$, $w\in V^*$, we set $|w|$ to be the length of the word w, $\#_a w$ to be the number of occurrences of a in w and $\#_{V'}w$ to be the number of occurrences of symbols of V' in w. The empty word is written as ε.

We assume that the reader is familiar with the fundamental definitions of formal language theory (e.g. see [9]). We recall the notion of a simple EG system as given in [5]. A *simple EG system* with n agents, $n\in\mathbb{N}$, is a construct

$$\Sigma=(V_E,P_E,R_1,\dots,R_n,\omega),$$

where (V_E,P_E,ω) is a 0L *system* with alphabet V_E, a set P_E of rewriting rules or productions $a\to v$ with $a\in V_E$, $w\in V_E^*$ such that for each $a\in V_E$ there exists a rule $a\to v$ in P_E (i.e., P_E is complete), and ω is the *axiom*. $R_1,\dots,R_n$ are sets of context-free rules or productions $a \to v$ with $a \in V_E$, $w \in V_E^*$. V_E is the set of symbols describing the environment, P_E is the set of *developmental rules of the environment* and every R_i, $i \in \{1,\dots,n\}$, is the set of *action rules* of the ith *agent*. A word $w \in V_E^*$ is also called a *state of the environment*. Note that the sets R_i which describe the agents, are not necessarily complete.

A simple EG system works in such a manner that it changes its states of environment according to the applications of the action rules of the agents and the developmental rules of the environment. More exactly, let $u, v \in V_E^*$. We say that u directly derives v in Σ according to the $=k$-team mode, $k \in \{0,\dots,n\}$, written as

$$u \overset{=k}{\Longrightarrow}_{\Sigma} v,$$

if $u = x_0 A_1 x_1 \dots x_{k-1} A_k x_k$, $v = y_0 \alpha_1 y_1 \dots y_{k-1} \alpha_k y_k$ with $A_i \in V_E$, $x_j, y_j, \alpha_i \in V_E^*$, $i \in \{1,\dots,k\}$, $j \in \{0,\dots,k\}$ and furthermore,

$$x_j \Longrightarrow_{P_E} y_j,\ j \in \{0,\dots,k\} \quad \text{(a derivation according to the 0L system of } \Sigma)$$

and

$$A_i \to \alpha_i \in R_{l_i}, l_i \in \{1,\dots,n\},\ l_j \neq l_m \quad \text{for } j \neq m, j, m \in \{1,\dots,k\}.$$

By the definition it is clear that $u \overset{=0}{\Longrightarrow}_{\Sigma} v$ equals the derivation $u \Longrightarrow_{P_E} v$ of the 0L system of Σ. In general, we see that exactly k of the n agents are chosen to be a *team* which, by applying exactly one of their action rules, replace exactly k symbols in the environmental state u while the other symbols are rewritten according to the 0L system of Σ, that is according to the developmental rules of the environment. We write

$$u \overset{\leqslant k}{\Longrightarrow}_{\Sigma} v$$

if $u \overset{=j}{\Longrightarrow}_{\Sigma} v$ for some $j \in \{0,\dots,k\}$. The transitive and reflexive closure of $\overset{k}{\Longrightarrow}_{\Sigma}$ or $\overset{\leqslant k}{\Longrightarrow}_{\Sigma}$ is denoted by $\overset{k}{\Longrightarrow}_{\Sigma}^{*}$ or $\overset{\leqslant k}{\Longrightarrow}_{\Sigma}^{*}$, respectively. Σ can be omitted if no confusion is possible. Let $e \in \{=k, \leqslant k \mid 0 \leqslant k \leqslant n\}$. We call

$$L(\Sigma, e) = \{v \mid v \in V_E^*,\ \omega \overset{e}{\Longrightarrow}_{\Sigma}^{*} v\}$$

the *language generated by Σ according to the e-team mode.*

Now we introduce the new concept of function-dependent teams. Let $\Sigma = (V_E, P_E, R_1, \dots, R_n, \omega)$ be a simple EG system and $f : \mathbb{N} \to \{0,\dots,n\}$ a mapping. Then

$$L(\Sigma, f) = \{v \mid \omega \overset{=f(1)}{\Longrightarrow} w_1 \overset{=f(2)}{\Longrightarrow} w_2 \overset{=f(3)}{\Longrightarrow} \cdots \overset{=f(r)}{\Longrightarrow} w_r = v, r \in \mathbb{N}_0, w_1, \dots, w_r \in V_E^*\}$$

is the *language generated by Σ according to the f-team mode*. We see that in the nth step of a derivation leading to a word $v \in L(\Sigma, f)$, exactly $f(n)$ agents are active. Furthermore,

$$L(\Sigma, \leqslant f) = \{v \mid \omega \overset{\leqslant f(1)}{\Longrightarrow} w_1 \overset{\leqslant f(2)}{\Longrightarrow} w_2 \overset{\leqslant f(3)}{\Longrightarrow} \cdots \overset{\leqslant f(r)}{\Longrightarrow} w_r = v, r \in \mathbb{N}_0, w_1, \dots, w_r \in V_E^*\}$$

is the *language generated by* Σ *according to the* $\leqslant f$*-team mode*. Obviously, for a constant function f with $f(n)=k$ for all $n\in\mathbb{N}$, we get the languages $L(\Sigma,=k)$ and $L(\Sigma,\leqslant k)$ as defined before.

We say that the agent R_i, $i\in\{1,\dots,n\}$, is *useful* if there exists a word $v\in L(\Sigma,f)$ (or $v\in L(\Sigma,\leqslant f)$) such that R_i is a member of some team used in a derivation from ω to v according to the f-team mode (or $\leqslant f$-team-mode, respectively). A simple EG system is *reduced* if all its agents are useful. By

$$\mathscr{E}\mathscr{L}(n,f) \quad\text{or}\quad \mathscr{E}\mathscr{L}(n,\leqslant f)$$

we denote the *family of languages generated by reduced simple EG systems with n agents according to the* f*-team mode or* $\leqslant f$*-team mode*, *respectively*.

Since the function $f:\mathbb{N}\to\{0,\dots,n\}$ is arbitrary it is also possible that f is not computable. In this case, $L(\Sigma,f)$ might be a language which is not recursively enumerable, but this is not always true. We shall discuss different possibilities after the following example.

If $w_1,\dots,w_m\in V_E^*$, $m\in\mathbb{N}$, then by $\mathrm{perm}(w_1,\dots,w_m)$ we denote the set of all concatenations of any permutation of $w_1,\dots,w_m$.

Example 2.1. Let $m,n\in\mathbb{N}$. We consider the simple *EG*-grammar system (with $n\geqslant 1$ agents)

$$\Sigma=(V_E,P_E,R_1,\dots,R_n,\omega),$$

where

$$\begin{aligned}
&V_E=\{a,b,b_1,\dots,b_n\},\\
&P_E=\{a\to a^2, b\to b^2\}\cup\{b_i\to b^2\mid i=1,\dots,n\},\\
&R_i=\{b\to bb_i\},\quad 1\leqslant i\leqslant n,\\
&\omega=a^2b^{2n+3m}.
\end{aligned}$$

The $3m$ supplementary occurrences of b in ω are introduced because of the proof of Theorem 3.1. Let $f:\mathbb{N}\to\{0,1,\dots,n\}$ be an arbitrary function. We consider the language $L(\Sigma,f)$ generated by Σ according to the f-team mode. A first derivation step is given by

$$a^2b^{2n+3m}\overset{=f(1)}{\Longrightarrow} w_1=a^4u_1 \text{ with an arbitrary}$$
$$u_1\in \mathrm{perm}(bb_{i_1},\dots,bb_{i_{f(1)}},\underbrace{b^2,\dots\dots\dots,b^2}_{(2n+3m)-f(1)\ \text{times}}),$$

where $1\leqslant i_1,\dots,i_{f(1)}\leqslant n$, $i_j\neq i_{j'}$ for $j\neq j'$, $1\leqslant j,j'\leqslant f(1)$. If in the next derivation step, we want to apply some agents R_i in $f(2)$ fixed, but arbitrary positions, then indeed, there exists a word generated by a first derivation step with the necessary occurrences of b at just these positions. This is true since there are at least n occurrences of b^2 in the words w_1. Analogous arguments hold for further derivation steps.

Thus,

$$L(\Sigma, f) = \{a^2 b^{2n+3m}\} \cup \bigcup_{k\in\mathbb{N}} \left(\bigcup_{\substack{1\leqslant i_1,\ldots,i_{f(k)}\leqslant n \\ (i_j\neq i_{j'}, j\neq j', 1\leqslant j,j'\leqslant f(k))}} \{a^{2^{k+1}}\}\mathrm{perm}(bb_{i_1},\ldots,bb_{i_{f(k)}},\underbrace{b^2,\ldots\ldots\ldots\ldots,b^2}_{2^{k-1}(2n+3m)-f(k)\ \text{times}})\right).$$

If $w_k,\ k\in\mathbb{N}_0$, is a word being derived in k steps according to the f-mode, then

$$\#_a w_k = 2^{k+1}, \quad \#_{\{b,b_1,\ldots,b_n\}} w_k = 2^k(2n+3m) \quad \text{and}$$
$$\#_{b_i} w_k \leqslant 1 \quad \text{for all } i\in\{1,\ldots,n\}.$$

It follows that for every $w\in L$, we get the length condition

$$2\cdot \#_{\{b,b_1,\ldots,b_n\}} w = (2n+3m)\cdot \#_a w.$$

Every word of $L(\Sigma, f)$ contains at least 2 occurrences of a and of b. We note that for the constant function $f(k)=0$ for all $k\in\mathbb{N}$, we get the 0L language $\{a^{2^{k+1}} b^{2^k(2n+3m)} | k\in\mathbb{N}_0\}$.

According to the $\leqslant f$-team mode, we generate the language

$$L(\Sigma, \leqslant f) = \{a^2 b^{2n+3m}\} \cup \bigcup_{k\in\mathbb{N}} \left(\bigcup_{\substack{1\leqslant i_1,\ldots,i_r\leqslant n, i_j\neq i_{j'}, j\neq j' \\ (1\leqslant j,j'\leqslant r, 0\leqslant r\leqslant f(k))}} \{a^{2^{k+1}}\}\,\mathrm{perm}(bb_{i_1},\ldots,bb_{i_r},\underbrace{b^2,\ldots\ldots\ldots\ldots,b^2}_{2^{k-1}(2n+3m)-r\ \text{times}})\right).$$

□

By Church's thesis, it is clear that for every simple EG system Σ and for every computable function $f:\mathbb{N}\to\{0,\ldots,n\}$, $L(\Sigma, f)$ and $L(\Sigma, \leqslant f)$ are recursively enumerable. For the languages of Example 2.1, we get more.

Theorem 2.1. *Let $f:\mathbb{N}\to\{0,\ldots,n\}$ be a function and $L=L(\Sigma, f)$ or $L=L(\Sigma,\leqslant f)$ the languages of Example* 2.1. *The function f is computable if and only if L is recursive.*

Proof. Assume that f is computable. Let $w\in V_E^*$. If $w=a^2b^{2n+3m}$, then $w\in L$. Else, it is decidable whether $w\in\{a^{2^{k+1}}\}\,\mathrm{perm}(bb_{i_1},\ldots,bb_{i_r},\underbrace{b^2,\ldots\ldots\ldots\ldots,b^2}_{2^{k-1}(2n+3m)-r\ \text{times}})$ for some $k,r\in\mathbb{N}$ and $1\leqslant i_1,\ldots,i_r\leqslant n$, $i_j\neq i_{j'}$ for $j\neq j'$ where $1\leqslant j,j'\leqslant r$ and $0\leqslant r\leqslant n$. We compute $f(k)$. In the case of $L=L(\Sigma,\leqslant f)$ it follows that $w\in L$ if and only if $r\leqslant f(k)$. In the case of $L=L(\Sigma, f)$ we have $w\in L$ if and only if $r=f(k)$. Thus, in any case, L is recursive.

For the other direction of the proof let L be recursive. Choose an arbitrary $k\in\mathbb{N}$. Consider the words $w_r=a^{2^{k+1}}bb_1\dots bb_rb^{2^k(2n+3m)-2r}$ for all $r\in\{0,\dots,n\}$. We know that at least one of these words belongs to L. Since L is recursive, for every r we can decide whether $w_r\in L$. We set $f(k)$ to be the maximum of all those r with $w_r\in L$. This shows that f is computable. □

For the language $L(\Sigma,f)$, we can prove even more.

Theorem 2.2. *Let $f:\mathbb{N}\to\{0,\dots,n\}$ be a function and $L=L(\Sigma,f)$ be the language of Example* 2.1. *If f is not computable then L is not recursively enumerable.*

Proof. Assume that L is recursively enumerable. This means that there exists an effective listing of all words of L. We choose an arbitrary $k\in\mathbb{N}$. There exists a word $w_k\in L$ with prefix $a^{2^{k+1}}b$, and this word is listed after a finite number of steps. We can compute $f(k)=\#_{\{b_1,\dots,b_n\}}w_k$. It follows that f is computable, a contradiction. □

The argument of the proof of Theorem 2.2 does not work for the language $L(\Sigma,\leqslant f)$. Obviously, we find a word $w_k\in L$ as above, but continuing the listing procedure, in the case of $\#_{\{b_1,\dots,b_n\}}w_k<n$, before not finding another word $w'_k\in L$ with the same prefix $a^{2^{k+1}}b$ and $\#_{\{b_1,\dots,b_n\}}w_k<\#_{\{b_1,\dots,b_n\}}w'_k$, we are never sure whether we will still find such a word.

For every recursively enumerable set $S\subseteqq\mathbb{N}$, we may consider its characteristic function $f:\mathbb{N}\to\{0,\dots,n\}$. If S is also recursive, then f is computable so that $L(\Sigma,\leqslant f)$ is recursive because of Theorem 2.1. If S is not recursive (see [8], p. 158 for an example of such a set), then we get the following result.

Theorem 2.3. *Let $S\subseteqq\mathbb{N}$ be a recursively enumerable set which is not recursive. Let $f:\mathbb{N}\to\{0,\dots,n\}$ be its characteristic function. Then $L=L(\Sigma,\leqslant f)$ of Example* 2.1 *is recursively enumerable, but not recursive.*

Proof. Since S is recursively enumerable, there exists an algorithm which lists all elements $s\in S$. From this algorithm, we construct an algorithm A_1 which for every listed $s\in S$ lists all words of the finite set

$$\bigcup_{i=1}^{n}\{a^{2^{s+1}}\}\,\mathrm{perm}(bb_i,\underbrace{b^2,\dots\dots\dots\dots,b^2}_{2^{s-1}(2n+3m)-1\ \text{times}}).$$

According to the definition of $L=L(\Sigma\leqslant f)$ in Example 2.1, this is the subset of L with $k=s$ and $f(s)=r=1$. Furthermore, there exists an algorithm A_2 which lists the remaining elements $a^{2^{k+1}}b^{2^k(2n+3m)}$ $(k=0,1,\dots)$ of L, i.e. the elements of the definition with $r=0$ for all $k\in\mathbb{N}_0$. From these two algorithm, we construct a new algorithm A which alternately lists an element according to A_1 and according to A_2 thus listing L. We conclude that L is recursively enumerable. Since f is not computable, L is not recursive. □

For fixed $n \in \mathbb{N}$, there exist languages which belong to all families $\mathscr{E}\mathscr{L}(n, f)$ and $\mathscr{E}\mathscr{L}(n, \leqslant f)$ for all functions $f : \mathbb{N} \to \{0, \ldots, n\}$. For example, for arbitrary $r \in \mathbb{N}$, taking the simple EG system $\Sigma' = (\{a\}, P_E, R_1, \ldots, R_n, a^r)$ with $P_E = R_1 = \cdots = R_n = \{a \to a\}$, it is clear that $\{a^r\} = L(\Sigma', f) = L(\Sigma', \leqslant f)$ for arbitrary functions $f : \mathbb{N} \to \{0, \ldots, n\}$.

3. More about $L(\Sigma, f)$ and $L(\Sigma, \leqslant f)$

The results of this section will be used in Section 4 to deliver comparability and incomparability results for different families of languages generated by EG systems with function-dependent teams.

Theorem 3.1. *Let $n, m \in \mathbb{N}$, and let $f : \mathbb{N} \to \{0, \ldots, n\}$ and $g : \mathbb{N} \to \{0, \ldots, m\}$ be functions such that there exist $k_1, k_2 \in \mathbb{N}$ with $f(k_1) > 1$ and $0 < f(k_2) < n$. If $m \neq n$ or $f \neq g$, then there exists a language $L \in \mathscr{E}\mathscr{L}(n, f)$ such that $L \notin \mathscr{E}\mathscr{L}(m, g)$.*

Proof. Let $L = L(\Sigma, f)$ be the language of Example 2.1. Let $\Sigma' = (V_E, P'_E, R'_1, \ldots, R'_m, \omega')$ be a simple EG system generating L according to the g-team mode.

We note that the m teams can only contribute a limited number of symbols to every derivation step. Because of the exponential growth of the lengths of the words of L, most of the occurrences of a and b must be generated by the help of productions $a \to w$ or $b \to v$ from the set P'_E of developmental rules of the environment. This implies that $w \in a^*$ and $v \in b^*$ since otherwise words with a mixture of occurrences of a and b or words with more than one occurrence of some b_i, $i = 1, \ldots, n$, could be generated.

For every $w \in L$, the length condition $2 \cdot \#_{\{b, b_1, \ldots, b_n\}} w = (2n + 3m) \cdot \#_a w$ is fulfilled. Assume that $a \to a^r$ and $a \to a^q$ are productions of P'_E, $r, q \in \mathbb{N}_0$. If $r \neq q$ and $a \to a^r$ is applied in a derivation step according to Σ', then we may exchange exactly one of its applications by an application of $a \to a^q$. After this alteration, we derive a word obviously not fulfilling the length condition. We conclude that $r = q$. Analogously, there exists exactly one production $b \to b^s$ in P'_E for some $s \in \mathbb{N}_0$. Because of the exponential growth of the lengths of the words, it follows that $r, s > 1$. Furthermore, $r = s$ since otherwise there could be generated words not fulfilling the length condition above.

Since there are at least 2 occurrences of a in every word of L, it follows that if there exists a production $a \to w$ in some R'_j, $j = 1, \ldots, m$, then $w \in a^*$. Analogously, if $b \to u$ or $b_i \to u$ for some agent R'_j, then $u \in \{b, b_1, \ldots, b_n\}^*$.

Consider any word $v \in L$. Then $|v| \geqslant 2 + 2n + 3m$. By any derivation step according to the g-team mode, at least $2 + 2n + 2m$ occurrences of a or of b are substituted by r new symbols using the productions $a \to a^r$ and $b \to b^r$ of P'_E. Obviously, this leads to a word v' with $|v'| \geqslant r(2 + 2n + 2m)$. Since $r > 1$ it follows that $|v'| > |a^2 b^{2n+3m}|$. We conclude that $a^3 b^{2n+3m}$ is the axiom of Σ'. If $r \geqslant 3$, then $|v'| \geqslant 6 + 6n + 6m$ which is longer than the second shortest words of L. This implies that $r = 2$. A nondeterministic behaviour of the system, if at all, is only possible by means of the agents.

Let $k' \in \mathbb{N}$ be the smallest number with $f(k')>0$. Then all the elements of the set

$$F_{k'} = \bigcup_{\substack{1 \leqslant i_1,\dots,i_{f(k')} \leqslant n \\ i_j \neq i_{j'}, j \neq j', 1 \leqslant j,j' \leqslant f(k')}} \left(\{a^{2^{k'+1}}\} \operatorname{perm}(bb_{i_1},\dots,bb_{i_{f(k')}}, \underbrace{b^2,\dots\dots\dots\dots,b^2}_{2^{k'-1}(2n+3m)-f(k') \text{ times}}) \right)$$

are derived according to the g-team mode from words without occurrences of b_i. Every b_i, $i=1,\dots,n$, is contained in at least one of the elements of this set. This implies that for every $i \in \{1,\dots,n\}$, there exists an agent R'_{z_i} containing a production $b \to \alpha_i b_i \beta_i$ for some $\alpha_i, \beta_i \in (V_E^* - \{a\})^*$, $j' \in \{1,\dots,n\}$.

Consider the word $w_{k_1} = a^{2^{k_1+1}} bb_1 b^{2^{k_1}(2n+3m)-2f(k_1)} bb_2 \dots bb_{f(k_1)} \in L$. This word is generated from some word $\bar{w} \in L$ according to Σ' in a step k''. Assume that $g(k'')=0$. Then the productions $a \to a^2$ and $b \to b^2$ have to be used. In the step k'', the subword ab of $\bar{w}$ (ab is a subword of every word of L!) would generate the subword a^2b^2 of w_{k_1}, a contradiction. It follows that $g(k'')>0$. But if for $g(k'')>0$ we would use a production $b_j \to \alpha_i b_i \beta_i$ of P'_E in step k'' and activate an agent R'_{z_i}, then we could generate a word with two occurrences of b_i which is not possible. Thus, a production $b_j \to \alpha_i b_i \beta_i$ of P'_E can only exist if it cannot be used for steps k with $g(k)>0$. Therefore, the case $g(k'')=1$ can only occur if b_1 and $b_{f(k_1)}$ are generated by a production of an agent with $b_1 b^{2^{k_1}(2n+3m)-2f(k_1)} bb_2 \dots bb_{f(k_1)}$ as subword of its right side. This contradicts the fact that in every derivation step at least m occurrences of b are substituted, according to P'_E, by b^2. This implies that $g(k'')>1$.

Assume that there exist $i,j \in \{1,\dots,n\}$ with $z_i \neq z_j$ such that $b \to \alpha_i b_i \beta_i$ is contained in both R'_{z_i} and R'_{z_j}. Then we consider $k'' \in \mathbb{N}$ with $g(k'')>1$. In a k''th derivation step according to Σ', we can apply both the agents R'_{z_i} and R'_{z_j} producing a word with 2 occurrences of b_i, a contradiction. We conclude that there exist n different agents R_{z_i} each of them containing a production $b \to \alpha_i b_i \beta_i$ for $i=1,\dots,n$. This implies that $m \geqslant n$. Then it is also clear that productions $b \to \alpha_i b_i \beta_i$ and $b \to \alpha_j b_j \beta_j$, $i \neq j$, cannot belong to the same agent R_{z_i}. By the same reasons, a production $b \to \alpha b_i \beta b_j \gamma$ for any $i,j \in \{1,\dots,n\}$ cannot exist.

Consider a production $b \to \alpha_i b_i \beta_i$ of R'_{z_i}. If there is an agent with a production $b_j \to \alpha'_i b_i \beta'_i$, $\alpha'_i, \beta'_i \in \{b, b_1, \dots, b_n\}$ which is applied at some derivation step according to Σ', then we may replace its application with the application of $b \to \alpha_i b_i \beta_i$ of R'_{z_i} which implies, by the length condition of the words, that $|\alpha_i b_i \beta_i| = |\alpha'_i b_i \beta'_i|$ (if these productions can be applied in a step k'' with $g(k'')>1$, then it follows that they belong to the same agent). Since it must be possible to generate a word of the set $F_{k'}$ above with abb_i as a subword (a subword ab_i does not occur!) there must exist a production $b \to bb_i \beta_i$ of R'_{z_i}. But since there is also a word with bb_i as a suffix in $F_{k'}$, there must be also a production $b \to \alpha_i bb_i$ in the agent R'_{z_i}. It follows that $|\alpha_i| = |\beta_i|$. Because of $g(k'')>1$ it is also possible to generate a word with suffix $bb_j bb_i$ for $i,j \in \{1,\dots,n\}$, $i \neq j$. We conclude that the only production in R'_{z_i} with left side b and an occurrence of b_i on its right side is $b \to bb_i$. If there exists some agent with a production with left side b_j and an occurrence of b_i on its right side, then the production is $b_j \to bb_i$.

We consider the word $w_{k_2} = a^{2^{k_2+1}} bb_1 \dots bb_{f(k_2)} b^{2^{k_2}(2n+3m)-2f(k_2)} \in L$, where $0 < f(k_2) < n$. By the considerations above, w_{k_2} is derived according to Σ' by a step k'' with $g(k'') > 0$ from a word $v \in L$. Assume that there exists a production $a \to a^p$, $p \in \mathbb{N}_0$, for some agent R' of Σ'. First, we consider the case that the agent R' is active in the derivation step $v \overset{=g(k'')}{\Longrightarrow} w_{k_2}$. If R' uses $a \to a^p$, then we skip this production and activate an agent R'_{z_j}, $j > f(k_2)$ (also if $R' = R'_{z_j}$). Since we know that in R'_{z_j} there do not exist any productions with left side b or b_i and right side belonging to b^*, we can generate a word w' with more than $f(k_2)$ occurrences of some symbols b_i, but with $\#_{\{b,b_1,\dots,b_n\}} w' = 2^{k_2(2n+3m)}$, a contradiction to the length condition of the words. If R', although active, does not use $a \to a^p$, then there remain only the two cases $R' = R'_{z_i}$ for some $i \in \{1, \dots, f(k_2)\}$ and $R' \neq R'_{z_i}$ for all $i \in \{1, \dots, n\}$. In the first case, the application of the production $b \to bb_i$ or $b_j \to bb_i$ of R' is exchanged with the application of $a \to a^p$. We derive a word w'' with $f(k_2) - 1$ occurrences of some b_i, but with $\#_{\{b,b_1,\dots,b_n\}} w'' = 2^{k_2(2n+3m)}$, a contradiction. In the second case, there must exist some production $b \to b^q$ or $b_i \to b^{r_i}$, $q, r_i \in \mathbb{N}_0$, $i \in \{1, \dots, n\}$, in R'. By skipping this production and activating the agent R'_{z_n}, we get a word w''' with more than $f(k_2) + 1$ occurrences of some b_i, but with $\#_a w''' = 2^{k_2+1}$, a contradiction. If the agent R' is not used at all in the derivation step leading to w_{k_2}, then we activate it instead of an agent R'_{z_i}, $i \in \{1, \dots, f(k_2)\}$, a contradiction again. It follows that occurrences of a are always substituted by the production $a \to a^2$ of P'_E. We conclude that every word of L is generated, whether according to Σ or to Σ', with the same number of derivation steps. Furthermore, if $f(k) > 0$, then $g(k) > 0$, too.

We consider the derivation step $v \overset{=g(k_2)}{\Longrightarrow} w_{k_2} = a^{2^{k_2+1}} bb_1 \dots bb_{f(k_2)} b^{2^{k_2}(2n+3m)-2f(k_2)}$ again. If $m > n$, then there must exist an agent $R' \neq R'_{z_i}$, $i \in \{1, \dots, n\}$. Therefore, R' must contain productions of the form $b \to b^q$ or $b_i \to b^{r_i}$. R' may be active or not in the derivation step above. In both cases we get similar contradictions as before. We conclude that $m = n$.

If for some $k \in \mathbb{N}_0$ we have $g(k) < f(k)$ then by Σ' we cannot generate the necessary $f(k)$ occurrences of b_i in $w_k = a^{2^{k+1}} bb_1 \dots bb_{f(k)} b^{2^k(2n+3m)-2f(k)}$. If $g(k) > f(k)$ (since $m = n$ this is only possible if $f(k) < n$), we can generate too much occurrences of symbols b_i. It follows that $f = g$. $\square$

For the language $L = L(\Sigma, f)$ considered in the proof of Theorem 3.1, the conditions imposed upon f are necessary. First, if there does not exist a $k_1 \in \mathbb{N}$ with $f(k_1) > 1$, then $f(k) \in \{0, 1\}$ for all $k \in \mathbb{N}$. We consider a function $g : \mathbb{N} \to \{0, \dots, m\}$ with $g(k) = f(k) \in \{0, 1\}$ for all $k \in \mathbb{N}$ and, with the notations of Example 2.1, for every $m \in \mathbb{N}$ we define the EG system $\Sigma' = (V_E, P_E, R'_1, \dots, R'_m, \omega)$ where $R'_m = \bigcup_{i=1}^n R_i$. Obviously, for all $m \in \mathbb{N}$, we have $L(\Sigma, f) = L(\Sigma', g)$. These considerations are also true for arbitrary languages $L \in \mathscr{EL}(n, f)$. We get

Proposition 3.1. *Let $n, m \in \mathbb{N}$, and let $f : \mathbb{N} \to \{0, \dots, n\}$ and $g : \mathbb{N} \to \{0, \dots, m\}$ be functions such that $f(k) = g(k) \in \{0, 1\}$ for all $k \in \mathbb{N}$. Then $\mathscr{EL}(n, f) = \mathscr{EL}(m, g)$.*

Second, if there does not exist $k_2 \in \mathbb{N}$ with $0 < f(k_2) < n$, then $f(k) \in \{0, n\}$ for all $k \in \mathbb{N}$. If $m > n$, we consider the function $g' : \mathbb{N} \to \{0, \dots, m\}$ where $g'(k) = m$ if $f(k) = n$ and $g'(k) = 0$ if $f(k) = 0$ and we define the EG system $\Sigma'' = (V_E, P_E, R_1, \dots, R_n, R'_{n+1}, \dots, R'_m, \omega)$ where $R'_{n+1} = \cdots = R'_m = P_E$. In this situation, $m \neq n$ and $f \neq g'$, but $L(\Sigma, f) = L(\Sigma'', g')$. These considerations are also true for every language $L \in \mathscr{E}\mathscr{L}(n, f)$ with $w = \varepsilon$ or $|w| \geqslant m$ for all $w \in L$.

Theorem 3.2. *Let $n, m \in \mathbb{N}$, and let $f : \mathbb{N} \to \{0, \dots, n\}$ and $g : \mathbb{N} \to \{0, \dots, m\}$ be functions such that there exists $k \in \mathbb{N}$ with $f(k) \geqslant 1$. Then there exists a language $L \in \mathscr{E}\mathscr{L}(n, f)$ such that $L \notin \mathscr{E}\mathscr{L}(m, \leqslant g)$.*

Proof. Let $L = L(\Sigma, f)$ be the language of Example 2.1. Assume that L is generated by an EG system according to the $\leqslant g$-team mode. Consider the proof of Theorem 3.1. The first steps of the proof leading to the axiom $a^2 b^{2n+3m}$ of Σ' and to the existence of productions $a \to a^2$ and $b \to b^2$ in P'_E are also valid if the g-team mode is substituted by the $\leqslant g$-team mode. Since $f(k) \geqslant 1$, all words of L with prefix $a^{2^{k+1}} b$ contain at least one occurrence of some b_i, $i \in \{1, \dots, n\}$. Thus, $a^{2^{k+1}} b^{2^k (2n+3m)} \notin L$. But the latter word can be derived according the $\leqslant g$-team mode, a contradiction. □

Theorem 3.3. *Let $n, m \in \mathbb{N}$, and let $f : \mathbb{N} \to \{0, \dots, n\}$ and $g : \mathbb{N} \to \{0, \dots, m\}$ be functions such that there exists $k_1 \in \mathbb{N}$ with $f(k_1) > 1$. Then there exists a language $L \in \mathscr{E}\mathscr{L}(n, \leqslant f)$ such that if $L \in \mathscr{E}\mathscr{L}(m, \leqslant g)$ then $m \geqslant n$ and furthermore, for $k_2 \in \mathbb{N}$, the following relations hold: if $f(k_2) < n$ then $f(k_2) = g(k_2)$ and if $f(k_2) = n$ then $g(k_2) \geqslant f(k_2)$.*

Proof. Let $L = L(\Sigma, \leqslant f)$ be the language of Example 2.1. Assume that L is generated by an EG system Σ' according to the $\leqslant g$-team mode. Consider the proof of Theorem 3.1 again. The steps of the proof leading to the axiom $a^2 b^{2n+3m}$ of Σ' and to the existence of productions $a \to a^2$ and $b \to b^2$ in P'_E are also valid in this case. If there is applied an agent in a derivation according to Σ' to some word v, then it is possible to apply exactly one agent or no agent to v, alternatively. Since $a \to a^2$ and $b \to b^2$ are productions in P'_E, it follows that the right side of the productions of an agent must have the length 2. This means that every word of L is generated by the same number of derivation steps, whether according to Σ or to Σ'. The considerations concerning the productions $b \to \alpha_i b i \beta_i$ of the agents can be carried over from the proof of Theorem 3.1 to this situation. Especially, since the right sides of the productions are of length 2, it is obvious that we get a production $b \to b b_i$ in the agent R'_{z_i}, $i = 1, \dots, n$. It is allowed that $a \to a^2$ or $b \to b^2$ belong to these agents, too. They are pairwise different so that $m \geqslant n$.

Assume that for some $k_2 \in \mathbb{N}$ we have $f(k_2) < n$. As in the proof of Theorem 3.1 we consider the word $w_{k_2} = a^{2^{k_2+1}} b b_1 \dots b b_{f(k_2)} b^{2^{k_2}(2n+3m) - 2f(k_2)}$. Obviously, $g(k_2) < f(k_2)$ and $g(k_2) > f(k_2)$ both lead to a contradiction. It follows that $f(k_2) = g(k_2)$.

If $f(k_2) = n$, then only $g(k_2) < f(k_2)$ delivers a contradiction. □

Let $m \geqslant n$, and let f and g be any functions fulfilling the conclusions of Theorem 3.3. An EG system $\Sigma' = (V_E, P'_E, R'_1, \ldots, R'_m, \omega')$ with $m \geqslant n$ agents generating $L(\Sigma, \leqslant f)$ according to the $\leqslant g$-team mode is given by

$$V'_E = \{a, b, b_1, \ldots, b_n\},$$
$$P'_E = \{a \to a^2, b \to b^2\} \cup \{b_i \to b^2 \mid i = 1, \ldots, n\},$$
$$R'_i = \{b \to bb_i\} \cup P'_E \quad \text{for } i = 1, \ldots, n,$$
$$R'_j = P'_E \quad \text{for } j = n+1, \ldots, m,$$
$$\omega = a^2 b^{2n+3m}.$$

The special case of a function f with $f(k) \in \{0, 1\}$ for all $k \in \mathbb{N}$ delivers the following.

Lemma 3.1. *Let $n, m \in \mathbb{N}$, and let $f : \mathbb{N} \to \{0, \ldots, n\}$ and $g : \mathbb{N} \to \{0, \ldots, m\}$ be functions with $f(k) \in \{0, 1\}$ for all $k \in \mathbb{N}$ such that there exists $k_1 \in \mathbb{N}$ with $f(k_1) = 1$. Then there exists a language $L \in \mathscr{E}\mathscr{L}(n, \leqslant f)$ such that if $L \in \mathscr{E}\mathscr{L}(m, \leqslant g)$ then the following relations hold: if $f(k_2) = 0$ then $g(k_2) = 0$ and if $f(k_2) = 1$ then $g(k_2) \geqslant f(k_2)$.*

Proof. As in the proof of Theorem 3.3 we get agents of Σ' with $b \to bb_i$, $i \in \{1, \ldots, n\}$. But these agents are not necessarily different so that we cannot conclude that $m \geqslant n$. But obviously, the two relations hold. $\square$

Let $f : \mathbb{N} \to \{0, \ldots, n\}$ and $g : \mathbb{N} \to \{0, \ldots, m\}$, $m \in \mathbb{N}$, be any functions fulfilling the conclusions of the lemma. Then the EG system $\Sigma' = (V_E, P'_E, R'_1, \ldots, R'_m, \omega)$ with

$$V'_E = \{a, b, b_1, \ldots, b_n\},$$
$$P'_E = \{a \to a^2, b \to b^2\} \cup \{b_i \to b^2 \mid i = 1, \ldots, n\},$$
$$R'_1 = \{b \to bb_1, \ldots, b \to bb_n\} \cup P'_E,$$
$$R'_j = P'_E \quad \text{for } j = 2, \ldots, m$$

and

$$\omega = a^2 b^{2n+3m}$$

generates $L(\Sigma, \leqslant f)$ according to the $\leqslant g$-team mode.

Theorem 3.4. *Let $n, m \in \mathbb{N}$, and let $f : \mathbb{N} \to \{0, \ldots, n\}$ and $g : \mathbb{N} \to \{0, \ldots, m\}$ be functions such that there exist $k_1, k_2 \in \mathbb{N}$ with $f(k_1) > 1$ and $0 < f(k_2) < n$. Then there exists a language $L \in \mathscr{E}\mathscr{L}(n, \leqslant f)$ such that if $L \in \mathscr{E}\mathscr{L}(m, g)$ then $m \geqslant n$ and furthermore, for $k_3 \in \mathbb{N}$, the following relations hold: if $f(k_3) < n$ then $f(k_3) = g(k_3)$ and if $f(k_3) = n$ then $g(k_3) \geqslant f(k_3)$.*

Proof. Let $L = L(\Sigma, \leqslant f)$ be the language of Example 2.1. Assume that L is generated by an EG system Σ' according to the g-team mode. The considerations of the proof of Theorem 3.1 leading to the axiom a^2b^{2n+3m}, to the unique productions $a \to a^2$ and $b \to b^2$ in P'_E and to the pairwise different agents R'_{z_i}, $i = 1, \dots, n$, each containing a production $b \to bb_i$ (and perhaps a production $b_j \to bb_i$), are also true in this case. It follows that $m \geqslant n$. If $a \to a^p$, $b \to b^q$ or $b_i \to b^{r_i}$, $i \in \{1, \dots, n\}$ (and $b_i \to b^{r_i}$ can be used in some derivation step) belong to an agent R'_{z_i}, then by the length condition of the words it is clear that $p = 2$, $q = 2$ or $r_i = 2$, respectively.

We consider the word $w_{k_2} = a^{2^{k_2+1}} bb_1 \dots bb_{f(k_2)} b^{2^{k_2}(2n+3m) - 2f(k_2)} \in L$. Let w_{k_2} be derived from a word $v \in L$ according to Σ' in step k'' with $g(k'') > 0$. We can assume that the agents R'_{z_j}, $j \in \{f(k_2+1), \dots, n\}$ are not active since otherwise we could also generate a word w' with more than $f(k_2)$ occurrences of some b_i, but with $\#_a w' = 2^{k_2+1}$. Assume that there exists an agent R', $R' \neq R'_{z_i}$, $i \in \{1, \dots, n\}$, containing a production $a \to a^p$, $p \in \mathbb{N}_0$. If R' is not used in this derivation step, then we can substitute an application of the production $b \to bb_1$ of the agent R'_{z_1} with an application of the production $a \to a^p$ in R'. We derive the word $a^{2^{k_2+1}+(p-2)} b^2 bb_2 \dots bb_{f(k_2)} b^{2^{k_2}(2n+3m)-2f(k_2)} \in L$. By the length condition of the words of L, it follows that $p = 2$. If R' is used in this derivation step, we deactivate R' and activate the agent $R'_{z_{f(k_2)+1}}$. Then we can generate more than $f(k_2)$ occurrences of some b_i while the total number of occurrences of symbols b and b_i remains the same, a contradiction. It follows that if a production $a \to a^p$ belongs to some agent, then $p = 2$.

This means that every occurrence of a is always substituted by a^2 irrespective of using P'_E or some agent (if possible). We conclude that every word of L is generated by the same number of derivation steps, whether according to Σ or to Σ'.

We consider the word $w_{k_3} = a^{2^{k_3+1}} bb_1 \dots bb_{f(k_3)} b^{2^{k_3}(2n+3m)-2f(k_3)}$ which is derived after k_3 derivation steps from v_{k_3}. If $g(k_3) < f(k_3)$, then w_{k_3} is not derivable according to Σ'. It follows that $f(k_3) \leqslant g(k_3)$. If $f(k_3) < n$ and $g(k_3) > f(k_3)$, then we can derive from v_{k_3} according to Σ' a word w'' with at least $f(k_3)+1$ occurrences of some b_i, but with $\#_a w'' = 2^{k_3+1}$, a contradiction. We conclude that if $f(k_3) < n$, then $g(k_3) = f(k_3)$. □

Again, we may consider the EG system Σ' defined after the proof of Theorem 3.3. It also generates $L(\Sigma, \leqslant f)$ according to the g-team mode. This is also true for the case that f does not fulfill the condition of Theorem 3.4, that is if $f(k) \in \{0, n\}$ for all $k \in \mathbb{N}$. But in the proof above, we could not exclude the case that $L(\Sigma, \leqslant f) \in \mathscr{EL}(m, g)$ for some function g not fulfilling the conclusions of Theorem 3.4.

4. Comparability results

First, we give a simple result which has already been stated in a similar form, that is for constant functions, in Theorem 4.1 of [5].

Theorem 4.1. (a) *$\mathscr{EL}(n, 0) = \mathscr{EL}(n, \leqslant 0) = \mathscr{L}(0\mathrm{L})$ for all $n \in \mathbb{N}_0$.*

(b) *Let $f : \mathbb{N} \to \{0, \dots, n\}$ be a function with $f(k) \in \{0, 1\}$ for all $k \in \mathbb{N}$ such that there exists $k_1 \in \mathbb{N}$ with $f(k_1) = 1$. Then $\mathscr{L}(0\mathrm{L}) \subsetneq \mathscr{EL}(n, f)$ and $\mathscr{L}(0\mathrm{L}) \subsetneq \mathscr{EL}(n, \leqslant f)$ for all $n \in \mathbb{N}$.*

Proof. Condition (a) is trivial.

For every 0L system $G=(V,P,\omega)$ we consider the EG system $\Sigma=(V,P,P_1,\ldots,P_n,\omega)$, where $P_1=\cdots=P_n=P$. Since at most one agent can be used, the derivations according to G or to Σ are the same. It follows that $L(G)=L(\Sigma,f)=L(\Sigma,\leqslant f)$. Since $\mathscr{L}(0\mathrm{L})=\mathscr{E}\mathscr{L}(n,0)=\mathscr{E}\mathscr{L}(n,\leqslant 0)$, the strict inclusions follows from Theorem 3.2 and Lemma 3.1. □

In [5], it is also proved that $\mathscr{L}(0\mathrm{L})$ and $\mathscr{E}\mathscr{L}(n,f)$ are incomparable for constant functions $f(k)=r$ for $r\geqslant 2$. For certain functions $f:\mathbb{N}\to\{0,\ldots,n\}$ such that there exists $k_1\in\mathbb{N}$ with $f(k_1)\geqslant 2$, we also can prove similar incomparability results but we could not prove them for arbitrary such functions. On the other side, if we define, as in Definition 4.1 in [5],

$$\mathscr{L}(0\mathrm{L},r)=\left\{L\mid L\in\mathscr{L}(0\mathrm{L}),L=\left(L-\bigcup_{i=1}^{r-1}(\mathrm{alph}\,L)^i\right)\right\}\quad\text{for } r\in\mathbb{N},$$

then we get the following:

Theorem 4.2. *Let $f:\mathbb{N}\to\{0,\ldots,n\}$, $n\in\mathbb{N}$, be a function such that there exists $k_1\in\mathbb{N}$ with $f(k_1)\geqslant 1$ and an $s\in\mathbb{N}$ with $f(k)\leqslant s\leqslant n$ for all $k\in\mathbb{N}$. If $r\geqslant s$, then $\mathscr{L}(0\mathrm{L},r)\subsetneq\mathscr{E}\mathscr{L}(n,f)$.*

Proof. It is obvious that $\mathscr{L}(0\mathrm{L},r+1)\subsetneq\mathscr{L}(0\mathrm{L},r)$ for all $r\in\mathbb{N}_0$. Thus, it suffices to prove $\mathscr{L}(0\mathrm{L},s)\subsetneqq\mathscr{E}\mathscr{L}(n,f)$. If G is a 0L system generating a language of $L\in\mathscr{L}(0\mathrm{L},s)$, then we can construct an EG system Σ as in the proof of Theorem 4.1. Since at most s agents can be used, Σ generates L according to the f-team mode. The strict inclusion is given by Theorem 3.2. □

For the $\leqslant f$-team mode, we can directly compare $\mathscr{E}\mathscr{L}(n,\leqslant f)$ with $\mathscr{L}(0\mathrm{L})$.

Theorem 4.3. *Let $f:\mathbb{N}\to\{0,\ldots,n\}$, $n\in\mathbb{N}$, be a function such that there exists $k_1\in\mathbb{N}$ with $1\leqslant f(k_1)\leqslant n$. Then $\mathscr{L}(0\mathrm{L})\subsetneqq\mathscr{E}\mathscr{L}(n,\leqslant f)$.*

Proof. Obviously, with an EG system as in the proof of Theorem 4.1 it follows that $\mathscr{L}(0\mathrm{L})\subseteqq\mathscr{E}\mathscr{L}(n,\leqslant f)$. The strict inclusion follows from Theorem 4.1(b) and, if $f(k_1)>1$, from Theorem 3.3 or, if $f(k)\in\{0,1\}$ for all $k\in\mathbb{N}$, from Lemma 3.1. □

Theorem 4.4. *Let $f:\mathbb{N}\to\{0,\ldots,m\}$, $m\in\mathbb{N}_0$, be a function such that there exists $k_1\in\mathbb{N}$ with $f(k_1)>1$, and let $n\in\mathbb{N}$ with $n>m$. Let $f':\mathbb{N}\to\{0,\ldots,n\}$ be the function with $f'(k)=f(k)$ for all $k\in\mathbb{N}$. Then $\mathscr{E}\mathscr{L}(m,\leqslant f)\subsetneqq\mathscr{E}\mathscr{L}(n,\leqslant f')$.*

Proof. Let $L\in\mathscr{E}\mathscr{L}(m,\leqslant f)$ be generated by an EG system $\Sigma=(V_E,P_E,P_1,\ldots,P_m,\omega)$ according to the f-team mode. Then we define the EG system $\Sigma'=(V_E,P_E,P_1,\ldots,P_m,P_{m+1},\ldots,P_n)$ where $P_{m+1}=\cdots=P_n=P_E$. Obviously, $L(\Sigma,\leqslant f)=L(\Sigma',\leqslant f')$. We conclude that $\mathscr{E}\mathscr{L}(m,\leqslant f)\subseteqq\mathscr{E}\mathscr{L}(n,\leqslant f')$. Since $n>m$ we know that $f(k_1)<n$. By

Theorem 3.3 there exists a language $L \in \mathscr{E}\mathscr{L}(n, \leqslant f')$ such that $L \notin \mathscr{E}\mathscr{L}(m, \leqslant f)$. This proves the strict inclusion. $\square$

Theorem 4.5. *Let $f : \mathbb{N} \to \{0, \ldots, m\}$ and $g : \mathbb{N} \to \{0, \ldots, n\}$ be functions such that there exist $k_1, k_1', k_2, k_2' \in \mathbb{N}$ with $f(k_1) > 1$, $g(k_1') > 1$, $g(k_2) \neq f(k_2) < m$ and $f(k_2') \neq g(k_2') < n$. Then the language families $\mathscr{E}\mathscr{L}(m, \leqslant f)$ and $\mathscr{E}\mathscr{L}(n, \leqslant g)$ are incomparable, but not disjoint.*

Proof. $\{a\}$ is a language belonging to both families. The incomparability result follows from Theorem 3.3. $\square$

Theorem 4.6. *Let $f : \mathbb{N} \to \{0, \ldots, m\}$ and $g : \mathbb{N} \to \{0, \ldots, n\}$ be functions such that there exist $k_1, k_1', k_2, k_2' \in \mathbb{N}$ with $f(k_1) > 1$, $g(k_1') > 1$, $1 < f(k_2) < m$ and $1 < g(k_2') < n$. If $m \neq n$ or $f \neq g$, then the language families $\mathscr{E}\mathscr{L}(m, f)$ and $\mathscr{E}\mathscr{L}(n, g)$ are incomparable, but not disjoint.*

Proof. $\{a\}$ is a language belonging to both families. The incomparability result follows from Theorem 3.1. $\square$

Theorem 4.7. *Let $n, m \in \mathbb{N}$, $f : \mathbb{N} \to \{0, \ldots, n\}$ and $g : \mathbb{N} \to \{0, \ldots, m\}$ functions such that there exist $k \in \mathbb{N}$ with $f(k) \geqslant 1$ and $k_1, k_2, k_3 \in \mathbb{N}$ with $g(k_1) > 1$, $0 < g(k_2) < n$, $g(k_3) < n$ and $f(k_3) \neq g(k_3)$. Then $\mathscr{E}\mathscr{L}(n, f)$ and $\mathscr{E}\mathscr{L}(m, \leqslant g)$ are incomparable, but not disjoint.*

Proof. $\{a\}$ is a language belonging to both families. By Theorem 3.2 (without using the condition imposed upon g) we know that there exists $L \in \mathscr{E}\mathscr{L}(n, f) - \mathscr{E}\mathscr{L}(m, \leqslant g)$. Theorem 3.4 proves that there exists a language $L' \in \mathscr{E}\mathscr{L}(m \leqslant g) - \mathscr{E}\mathscr{L}(n, f)$. $\square$

The incomparability of the language families $\mathscr{E}\mathscr{L}(n, f)$ and $\mathscr{E}\mathscr{L}(n, \leqslant g)$ is also true for more functions.

Theorem 4.8. *Let $n, m \in \mathbb{N}$, and let $f, g : \mathbb{N} \to \{0, \ldots, n\}$ be functions such that there exist $k \in \mathbb{N}$ with $f(1) > 1$. Then $\mathscr{E}\mathscr{L}(n, f)$ and $\mathscr{E}\mathscr{L}(n, \leqslant g)$ are incomparable, but not disjoint.*

Proof. We consider the language $L = \{a, b\}$ which is generated by the EG system $\Sigma = (\{a, b\}, P_E, R_1, \ldots, R_n, a)$ with $P_E = R_1 = \cdots = R_n = \{a \to b, b \to b\}$ according to the $\leqslant g$-team mode. But for $f(1) > 1$, L cannot be generated by any EG system according to the f-team mode. $\square$

One might suspect that a similar proof also works for functions f such there exists a $k \in \mathbb{N}$ with $f(k) > 1$, e.g. $f(1) = 1$, $f(2) = 2$ and $f(k) = 1$ for $k \geqslant 2$. But here, a first derivation step according to the f-mode is always possible. Therefore, in this case all languages only containing words of length 1 can be generated.

For certain functions we get strict inclusion results:

Theorem 4.9. *Let* $n, m \in \mathbb{N}$, *and let* $f : \mathbb{N} \to \{0, \dots, n\}$ *and* $f' : \mathbb{N} \to \{0, \dots, m\}$ *be functions such that* $f(k) = f'(k) \in \{0, 1\}$ *for all* $k \in \mathbb{N}$. *If there exist a* $k_1 \in \mathbb{N}$ *with* $f(k_1) = 1$, *then* $\mathscr{E}\mathscr{L}(n, \leqslant f) \subsetneq c\mathscr{E}\mathscr{L}(m, f')$.

Proof. Let $L = L(\Sigma, \leqslant f)$ for some EG system $\Sigma = (V_E, P_E, R_1, \dots, R_n, \omega)$. Then L is generated by the EG system $\Sigma' = (V_E, P_E, R_1 \cup P_E, \dots, R_n \cup P_E, R_{n+1}, \dots, R_m, \omega)$ with $R_{n+1} = \cdots = R_m = P_E$ according to the f'-team mode. By Theorem 3.2, the strict inclusion follows. □

Acknowledgements

The author likes to thank the anonymous referee for his helpful comments.

References

[1] M.H. ter Beek, Teams in grammar systems: hybridity and weak rewriting, Acta Cybern. 12 (1996) 427–444.
[2] M.H. ter Beek, Simple eco-grammar systems with prescribed teams, in: G. Păun, A. Salomaa (Eds.), Grammatical Models of Multi-agent Systems, Gordon and Breach, London, 1999, pp. 113–135.
[3] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenová, G. Păun, Eco-grammar systems, a preview, in: R. Trappl (Ed.), Proc. 12th European Meeting on Cybernetics and System Research, World Scientific Publishers, Singapore, 1994, pp. 941–948.
[4] E. Csuhaj-Varjú, J. Kelemen, A. Kelemenová, G. Păun, Eco-grammar systems, a grammatical framework for studying lifelike interactions, Artif. Life 3 (1997) 1–28.
[5] E. Csuhaj-Varjú, A. Kelemenová, Team behaviour in eco-grammar systems, Theoret. Comput. Sci. 209 (1998) 213–224.
[6] E. Csuhaj-Varjú, V. Mitrana, Dynamical teams in eco-grammar systems, Fund. Inform. 44 (2000) 83–94.
[7] H. Fernau, On function-limited Lindenmayer systems, J. Inform. Process. Cybernet. EIK 27 (1991) 21–53.
[8] A.J. Kfoury, R.N. Moll, M.A. Arbib, A Programming Approach to Computability, Springer, New York, 1982.
[9] A. Salomaa, Formal Languages, Academic Press, New York, 1973.
[10] D. Wätjen, D. Ostrovsky, Function-limited 0L systems revisited, J. Automata, Languages Combinat. 6 (2001) 97–114.